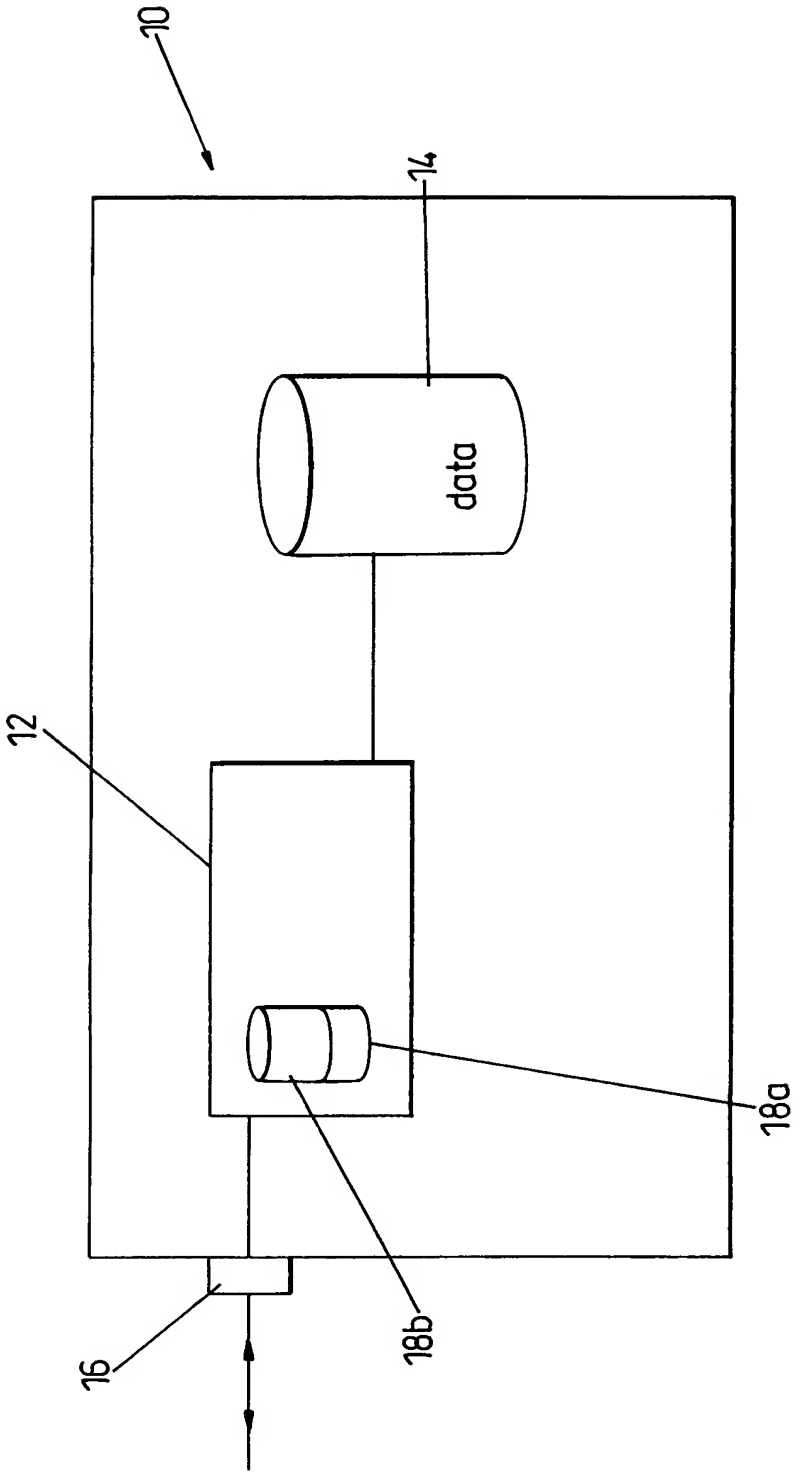


Fig. 1

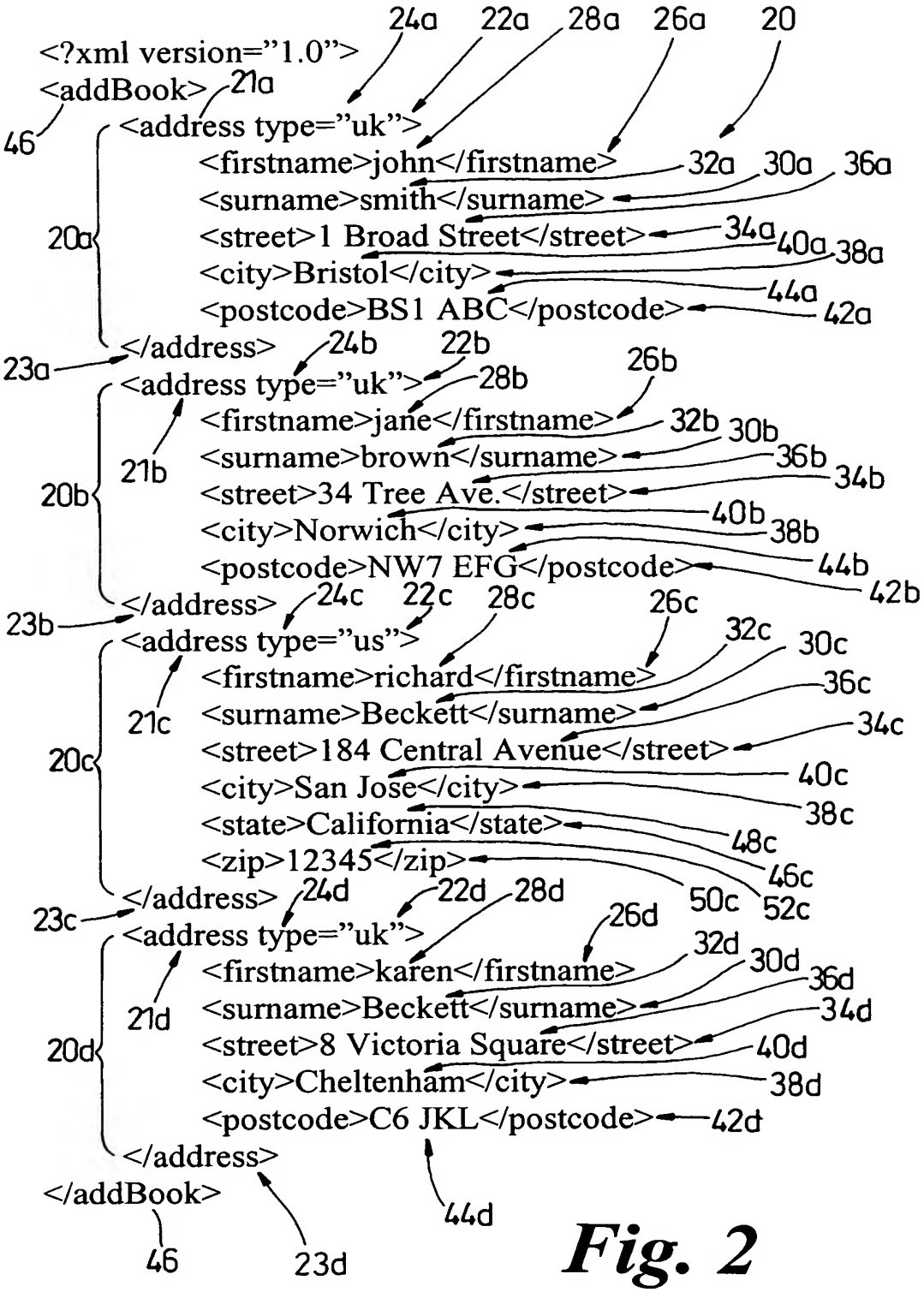
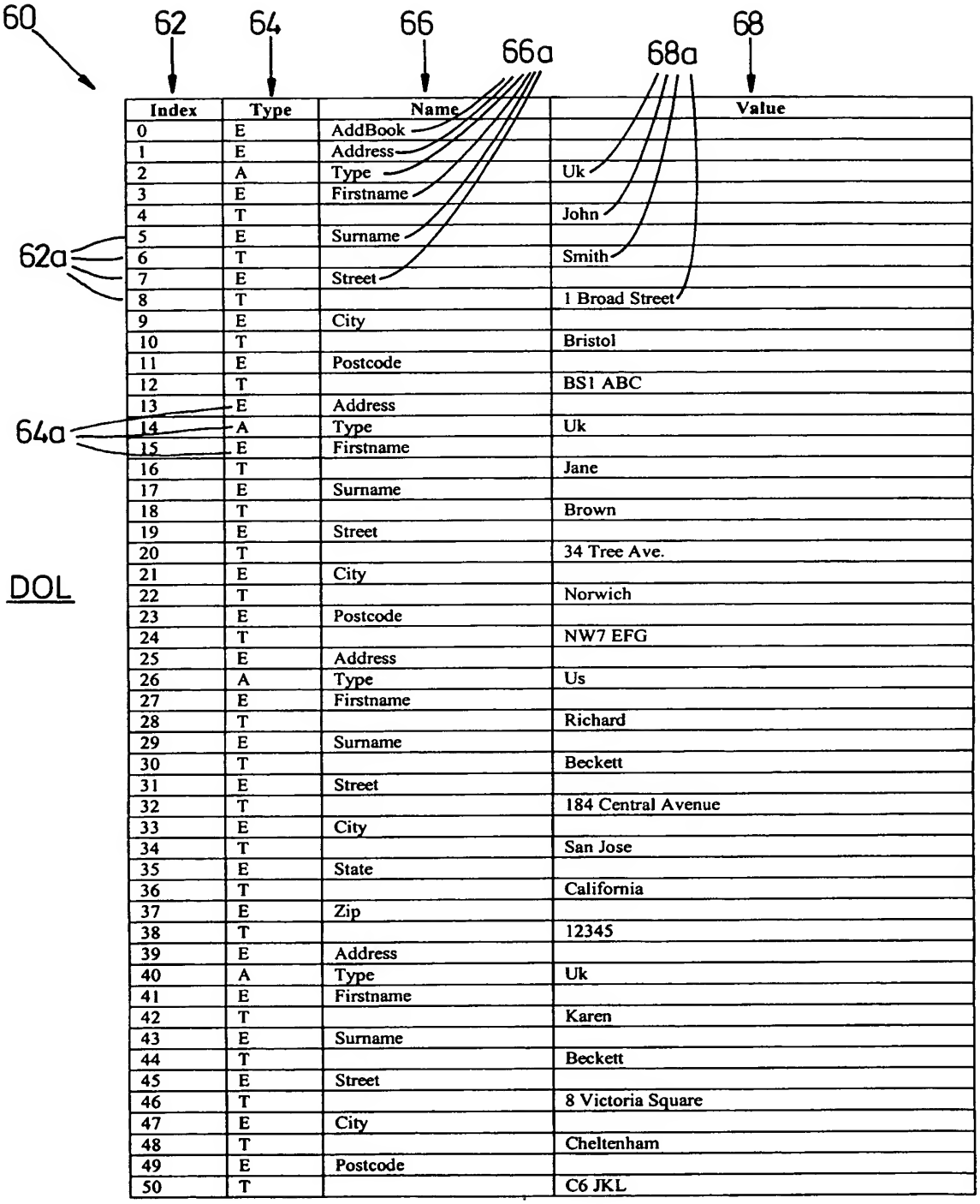
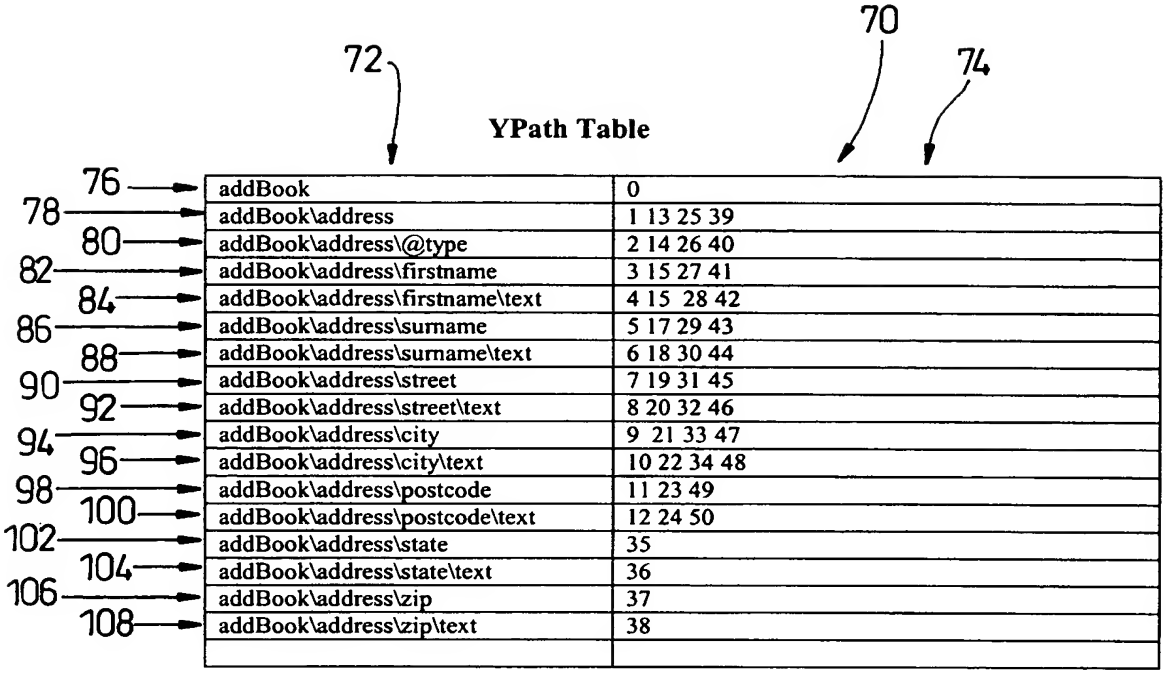

<?xml version="1.0"?>
<addBook>
<address type="uk">
<firstname>john</firstname>
<surname>smith</surname>
<street>1 Broad Street</street>
<city>Bristol</city>
<postcode>BS1 ABC</postcode>
</address>
<address type="uk">
<firstname>jane</firstname>
<surname>brown</surname>
<street>34 Tree Ave.</street>
<city>Norwich</city>
<postcode>NW7 EFG</postcode>
</address>
<address type="us">
<firstname>richard</firstname>
<surname>Beckett</surname>
<street>184 Central Avenue</street>
<city>San Jose</city>
<state>California</state>
<zip>12345</zip>
</address>
<address type="uk">
<firstname>karen</firstname>
<surname>Beckett</surname>
<street>8 Victoria Square</street>
<city>Cheltenham</city>
<postcode>C6 JKL</postcode>
</address>
</addBook>
46
21a
24a
22a
28a
26a
20
32a
30a
36a
20a
34a
40a
38a
44a
42a
23a
24b
22b
26b
28b
21b
32b
30b
36b
20b
34b
40b
38b
44b
42b
23b
24c
22c
28c
26c
21c
32c
30c
36c
20c
34c
40c
38c
48c
46c
50c
52c
23c
24d
22d
28d
26d
21d
32d
30d
36d
20d
34d
40d
38d
42d
44d
46
23d

Fig. 2



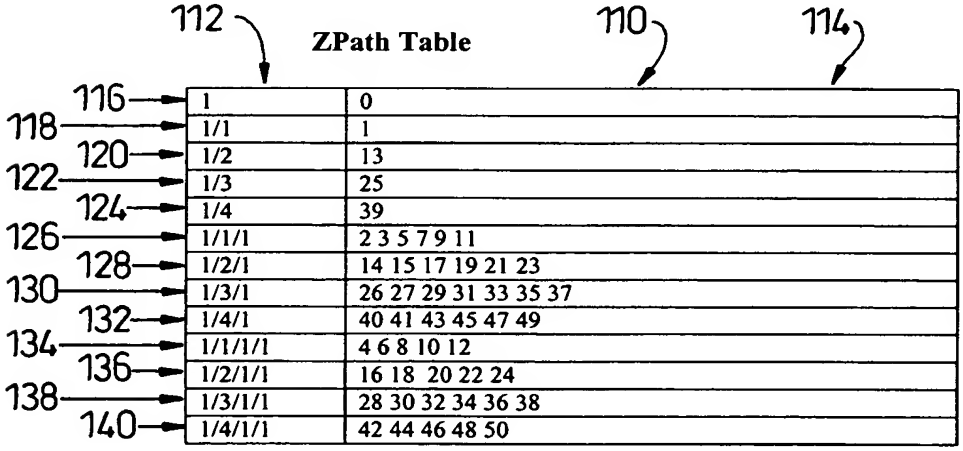
Index	Type	Name	Value
0	E	AddBook	
1	E	Address	
2	A	Type	Uk
3	E	Firstname	
4	T		John
5	E	Surname	
6	T		Smith
7	E	Street	
8	T		1 Broad Street
9	E	City	
10	T		Bristol
11	E	Postcode	
12	T		BS1 ABC
13	E	Address	
14	A	Type	Uk
15	E	Firstname	
16	T		Jane
17	E	Surname	
18	T		Brown
19	E	Street	
20	T		34 Tree Ave.
21	E	City	
22	T		Norwich
23	E	Postcode	
24	T		NW7 EFG
25	E	Address	
26	A	Type	Us
27	E	Firstname	
28	T		Richard
29	E	Surname	
30	T		Beckett
31	E	Street	
32	T		184 Central Avenue
33	E	City	
34	T		San Jose
35	E	State	
36	T		California
37	E	Zip	
38	T		12345
39	E	Address	
40	A	Type	Uk
41	E	Firstname	
42	T		Karen
43	E	Surname	
44	T		Beckett
45	E	Street	
46	T		8 Victoria Square
47	E	City	
48	T		Cheltenham
49	E	Postcode	
50	T		C6 JKL

Fig. 3



addBook	0
addBook\address	1 13 25 39
addBook\address\@type	2 14 26 40
addBook\address\firstname	3 15 27 41
addBook\address\firstname\text	4 15 28 42
addBook\address\surname	5 17 29 43
addBook\address\surname\text	6 18 30 44
addBook\address\street	7 19 31 45
addBook\address\street\text	8 20 32 46
addBook\address\city	9 21 33 47
addBook\address\city\text	10 22 34 48
addBook\address\postcode	11 23 49
addBook\address\postcode\text	12 24 50
addBook\address\state	35
addBook\address\state\text	36
addBook\address\zip	37
addBook\address\zip\text	38

Fig. 4



1	0
1/1	1
1/2	13
1/3	25
1/4	39
1/1/1	2 3 5 7 9 11
1/2/1	14 15 17 19 21 23
1/3/1	26 27 29 31 33 35 37
1/4/1	40 41 43 45 47 49
1/1/1/1	4 6 8 10 12
1/2/1/1	16 18 20 22 24
1/3/1/1	28 30 32 34 36 38
1/4/1/1	42 44 46 48 50

Fig 5

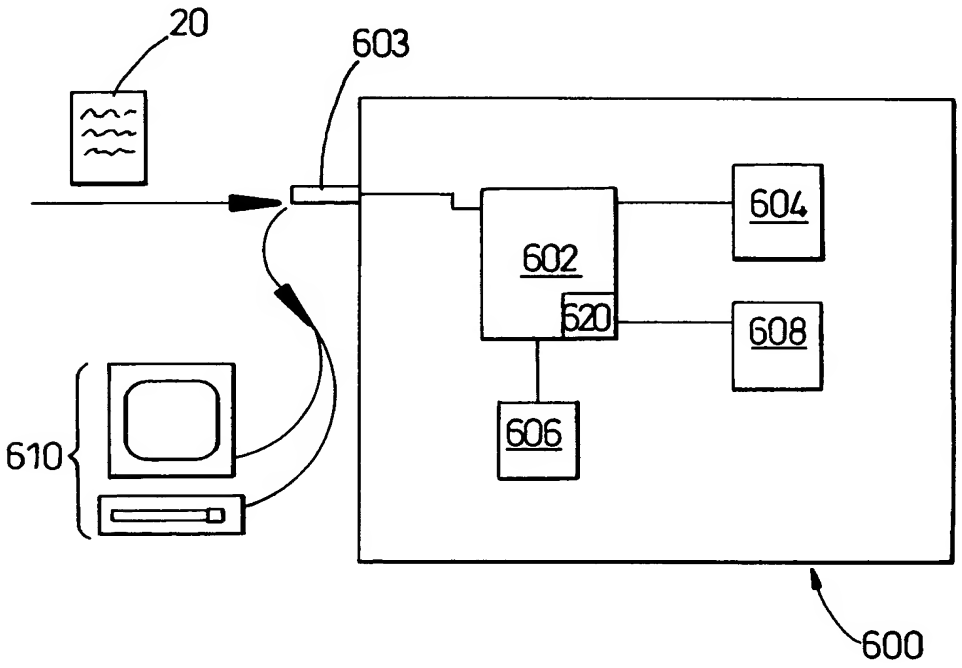


Fig. 6A

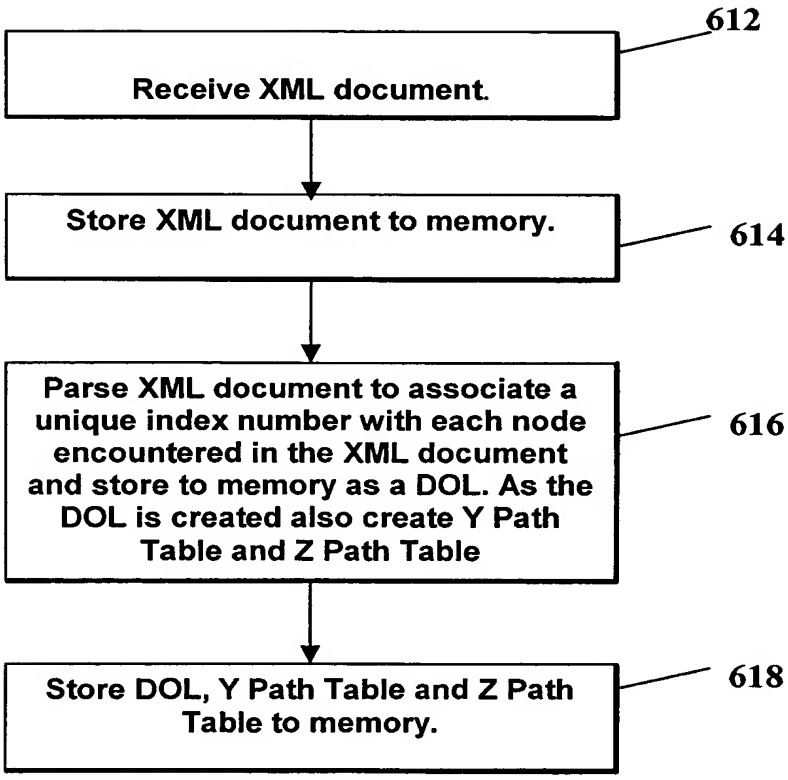

612
Receive XML document.
Store XML document to memory.
614
Parse XML document to associate a unique index number with each node encountered in the XML document and store to memory as a DOL. As the DOL is created also create Y Path Table and Z Path Table
616
Store DOL, Y Path Table and Z Path Table to memory.
618

Fig. 6B

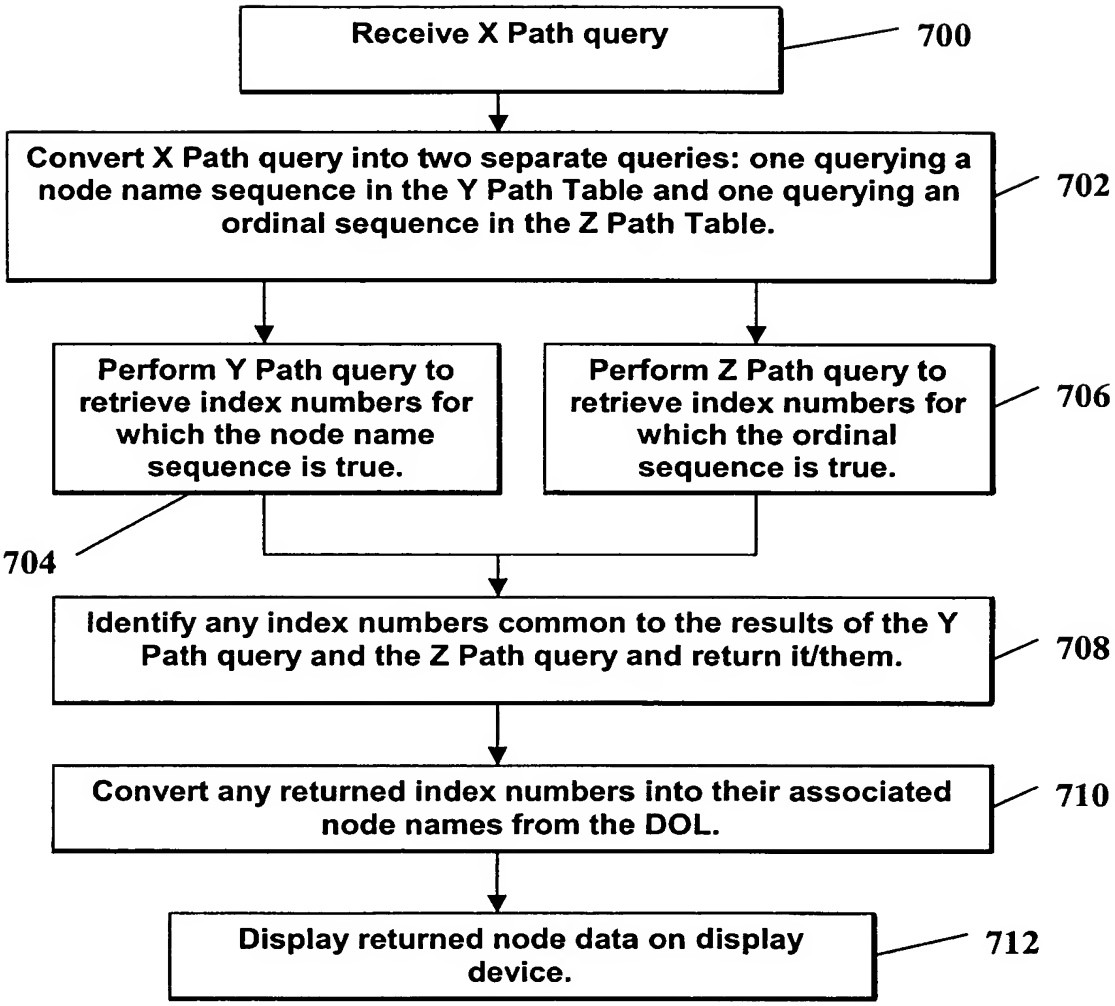

Receive X Path query
700
Convert X Path query into two separate queries: one querying a node name sequence in the Y Path Table and one querying an ordinal sequence in the Z Path Table.
702
Perform Y Path query to retrieve index numbers for which the node name sequence is true.
Perform Z Path query to retrieve index numbers for which the ordinal sequence is true.
706
704
Identify any index numbers common to the results of the Y Path query and the Z Path query and return it/them.
708
Convert any returned index numbers into their associated node names from the DOL.
710
Display returned node data on display device.
712

Fig. 7

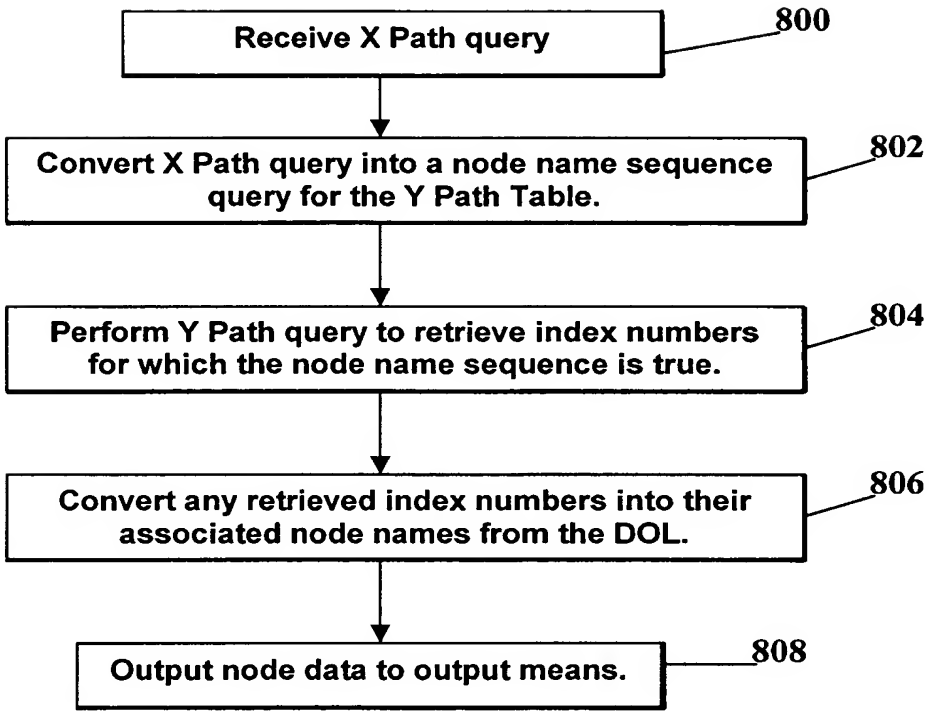

Receive X Path query
800
Convert X Path query into a node name sequence query for the Y Path Table.
802
Perform Y Path query to retrieve index numbers for which the node name sequence is true.
804
Convert any retrieved index numbers into their associated node names from the DOL.
806
Output node data to output means.
808

Fig. 8

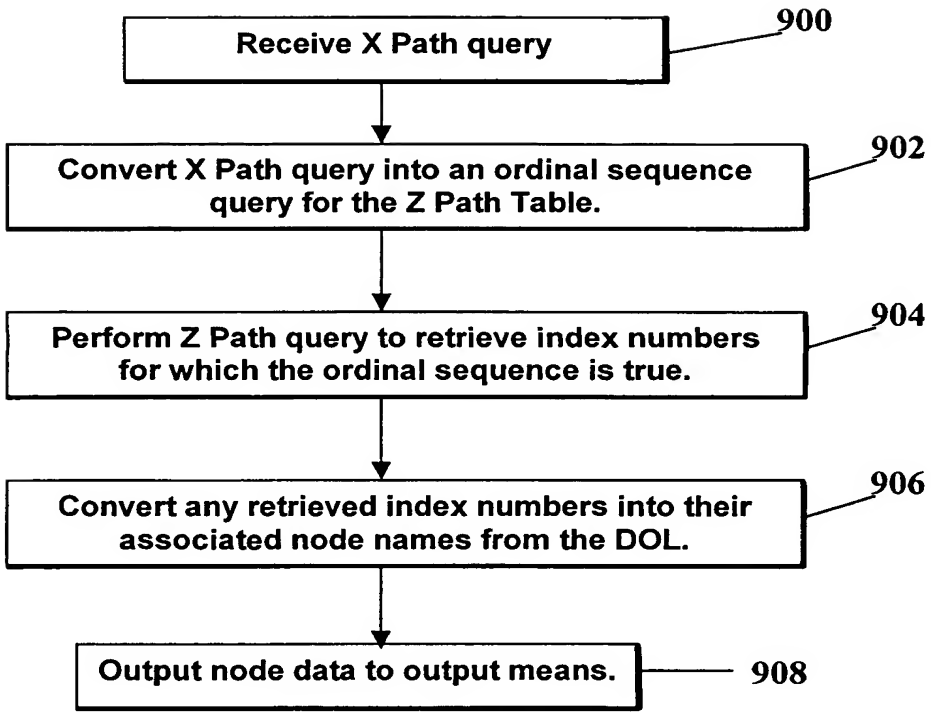

Receive X Path query
900
Convert X Path query into an ordinal sequence query for the Z Path Table.
902
Perform Z Path query to retrieve index numbers for which the ordinal sequence is true.
904
Convert any retrieved index numbers into their associated node names from the DOL.
906
Output node data to output means.
908

Fig. 9

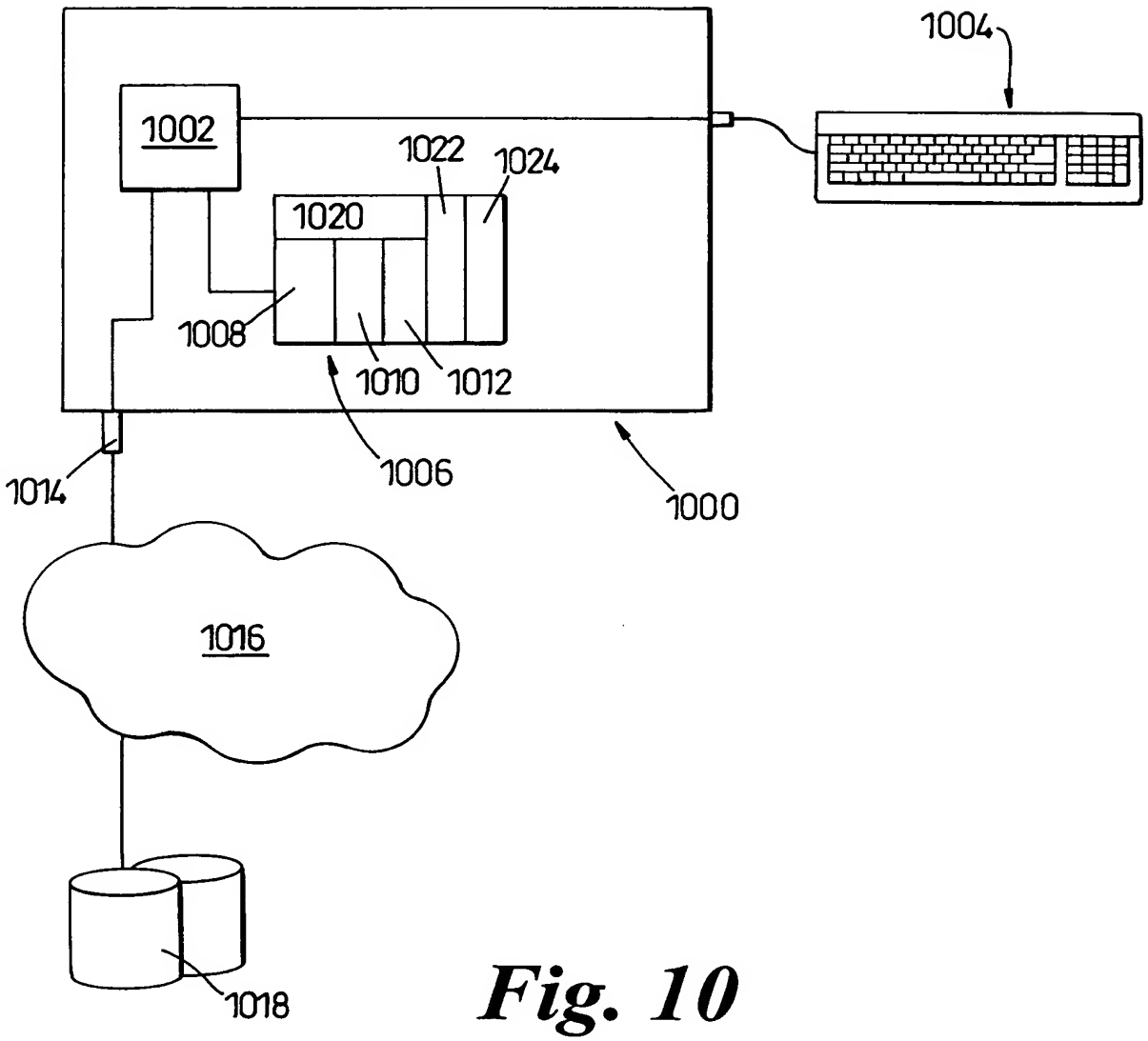

1004
1002
1022
1024
1020
1008
1010
1012
1014
1006
1000
1016
1018

Fig. 10

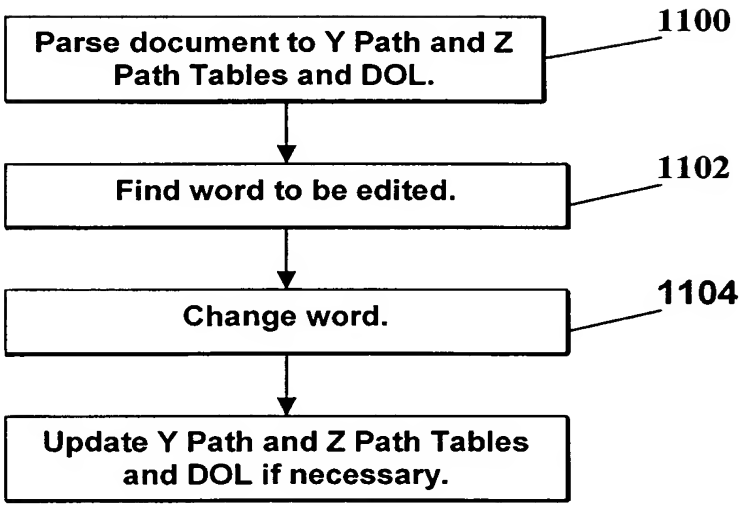

Parse document to Y Path and Z Path Tables and DOL.
1100
Find word to be edited.
1102
Change word.
1104
Update Y Path and Z Path Tables and DOL if necessary.

Fig. 11

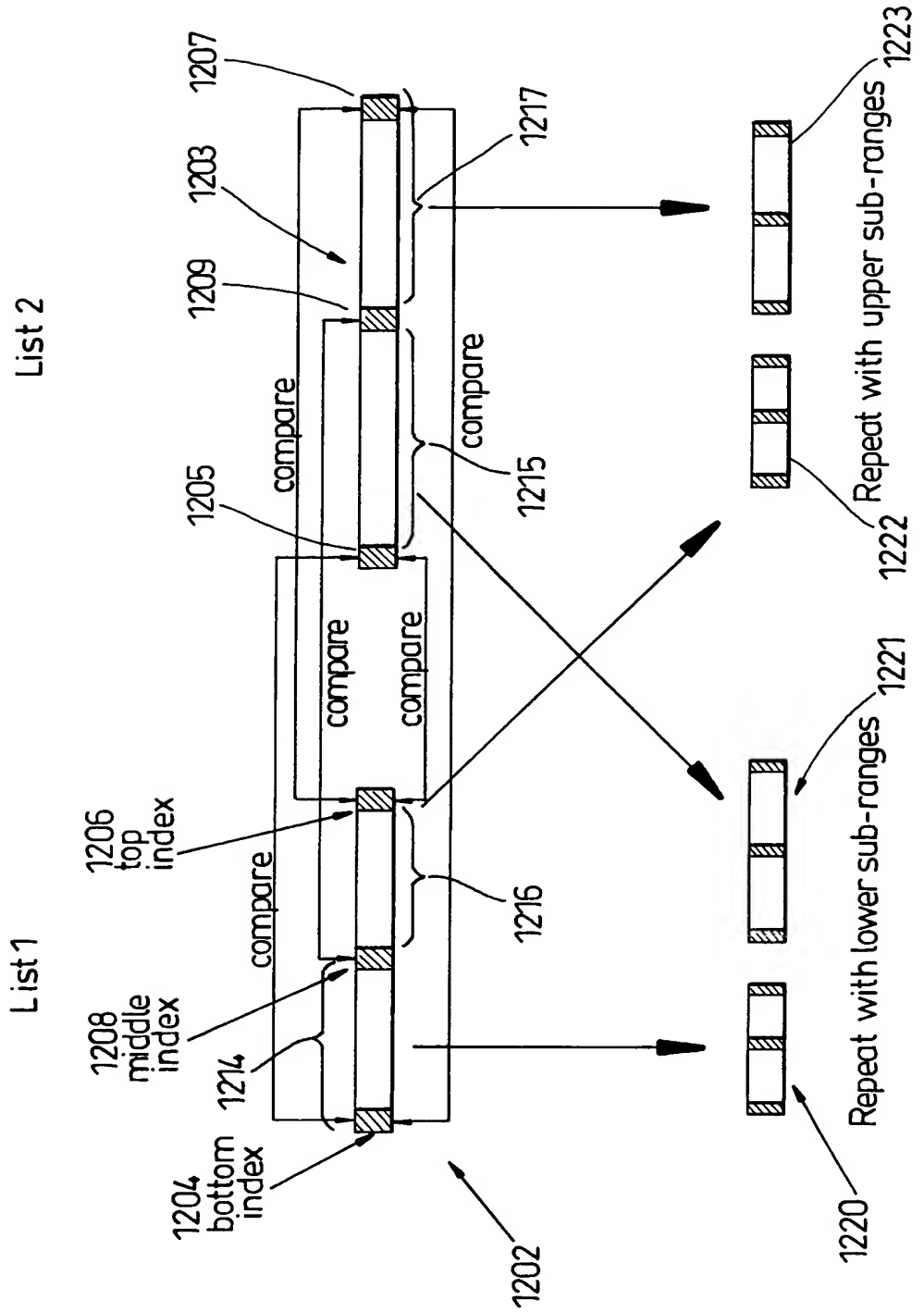

List 1
List 2
1202
1204 bottom index
1206 top index
1208 middle index
1214
1216
1217
1220
1221
1222
1223
1205
1207
1209
1215
compare
compare
compare
compare
Repeat with lower sub-ranges
Repeat with upper sub-ranges

Fig. 12

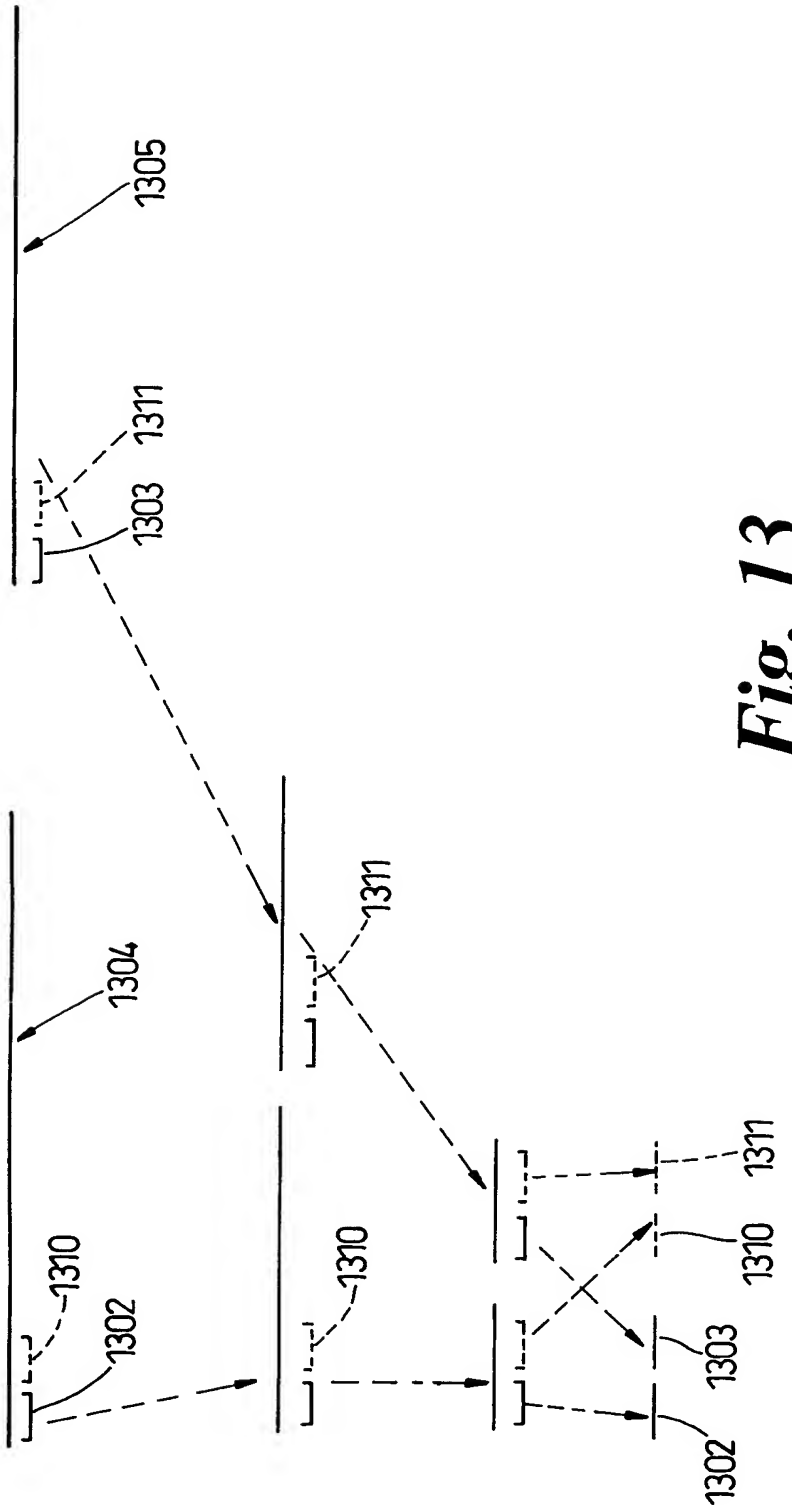


Fig. 13

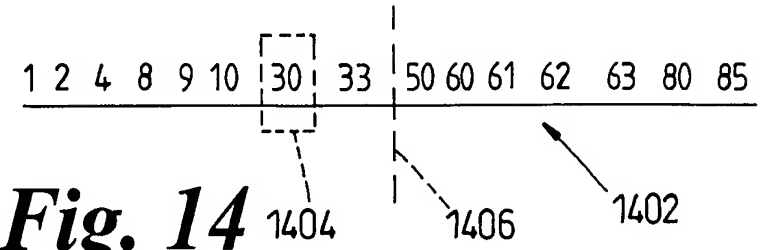
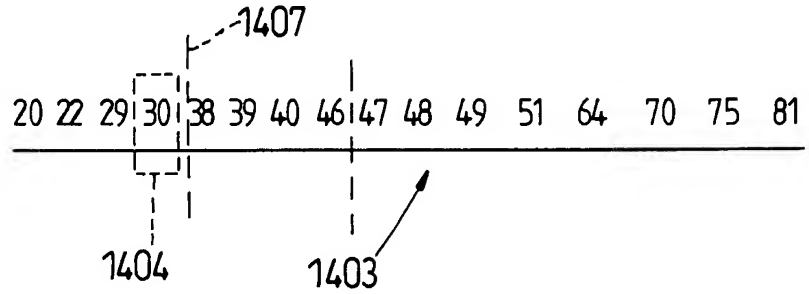

1 2 4 8 9 10 30 33 50 60 61 62 63 80 85
1404
1406
1402

Fig. 14


1407
20 22 29 30 38 39 40 46 47 48 49 51 64 70 75 81
1404
1403

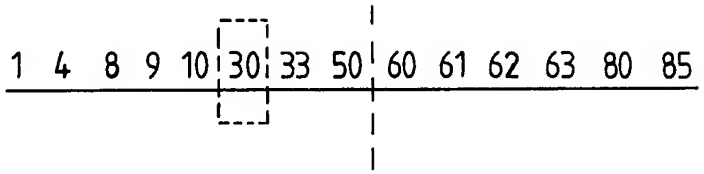
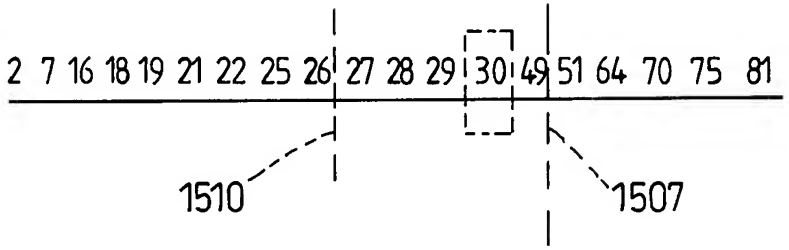

1 4 8 9 10 30 33 50 60 61 62 63 80 85

Fig. 15


2 7 16 18 19 21 22 25 26 27 28 29 30 49 51 64 70 75 81
1510
1507

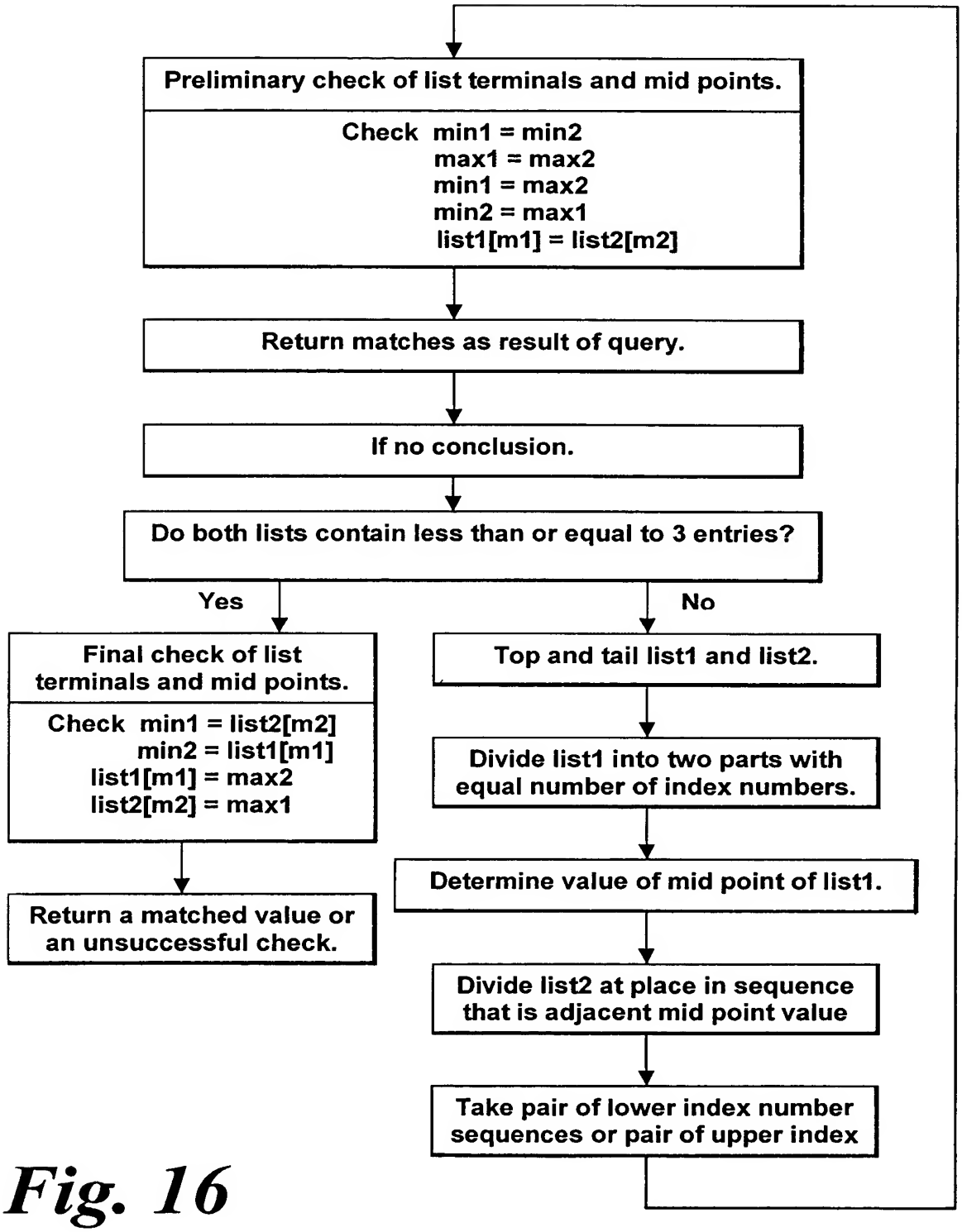

Preliminary check of list terminals and mid points.
Check min1 = min2
max1 = max2
min1 = max2
min2 = max1
list1[m1] = list2[m2]
Return matches as result of query.
If no conclusion.
Do both lists contain less than or equal to 3 entries?
Yes
No
Final check of list terminals and mid points.
Check min1 = list2[m2]
min2 = list1[m1]
list1[m1] = max2
list2[m2] = max1
Return a matched value or an unsuccessful check.
Top and tail list1 and list2.
Divide list1 into two parts with equal number of index numbers.
Determine value of mid point of list1.
Divide list2 at place in sequence that is adjacent mid point value
Take pair of lower index number sequences or pair of upper index

Fig. 16